UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/731,899	12/09/2003	Brian Jones	60001.0182USI1/MS303914.1	4706

7590 08/02/2006

Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
CHEN, QING	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 08/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/731,899	**Applicant(s)** JONES ET AL.	
	Examiner Qing Chen	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *09 December 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 1/23/04, 8/6/04, 5/26/05, 7/29/05, 11/2/05
4/24/06, 5/22/06, 7/3/06

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on December 9, 2003. **Claims 1-18** are currently pending and have been considered below.

Information Disclosure Statement

2. The information disclosure statements (IDS) filed on January 23, 2004, August 6, 2004, May 26, 2005, July 29, 2005, November 2, 2005, April 24, 2006, May 4, 2006, May 22, 2006, and July 3, 2006 have been received. The submissions on January 23, 2004, August 6, 2004, May 26, 2005, July 29, 2005, November 2, 2005, April 24, 2006, May 22, 2006, and July 3, 2006 are in compliance with the provisions of 37 CFR 1.97, and accordingly, these information disclosure statements have been considered by the Examiner.

The information disclosure statement filed on May 4, 2006 fails to comply with the provisions of 37 CFR 1.97, 1.98 and MPEP § 609 because it lacks the form PTO-1449 or PTO/SB/08A and 08B, "Information Disclosure Statement," as set forth in 37 CFR 1.98(a)(1). It has been placed in the application file, but the information referred to therein has not been considered as to the merits. Applicant is advised that the date of any re-submission of any item of information contained in this information disclosure statement or the submission of any missing element(s) will be the date of submission for purposes of determining compliance with the requirements based on the time of filing the statement, including all certification requirements for statements under 37 CFR 1.97(e). See MPEP § 609.05(a).

The information disclosure statements filed on January 23, 2004, November 2, 2005, and April 24, 2006 fail to comply with 37 CFR 1.98(a)(2), which requires a legible copy of each cited foreign patent document; each non-patent literature publication or that portion which caused it to be listed; and all other information or that portion which caused it to be listed. It has been placed in the application file, but the information referred to therein has not been considered.

Oath/Declaration

3. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:

- The clause regarding "willful false statements ..." required by 37 CFR 1.68 has been omitted.
- It does not identify the citizenship of one of the inventors.
- The full name of one of the inventors (family name and at least one given name together with any initial) has not been set forth.
- It does not identify the mailing address of one of the inventors. A mailing address is an address at which an inventor customarily receives his or her mail and may be either a home or business address. The mailing address should include the ZIP Code designation. The mailing address may be provided in an application data sheet or a supplemental oath or declaration. See 37 CFR 1.63(c) and 37 CFR 1.76.
- It does not specify the domestic priority information that of the application on which priority is claimed, by specifying the application number, filing date, and status.

Drawings

4. The drawings are objected to under 37 CFR 1.83(a) because they fail to show the manifest in the RAM (Figure 2, Element 25), the hard disk drive (Figure 2, Element 27), the magnetic disk drive (Figure 2, Element 28), and the optical disk drive (Figure 2, Element 30) as

described in the specification. Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "110" has been used to designate both "the document" and "the performance review." Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference characters not mentioned in the description:

- Element 105 in Figure 2;
- Elements 140 and 150 in Figure 3;
- Elements 500, 510, and 520 in Figure 5;
- Elements 610, 620, 630, and 640 in Figure 6;
- Element 710 in Figure 7;
- Element 810 in Figure 8;
- Element 910 in Figure 9; and
- Element 1120 in Figure 11.

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objections to the drawings will not be held in abeyance.

Specification

5. The disclosure is objected to because of the following informalities:

- The specification contains the following typographical errors:
    - o The letter "e" in "resume" should be replaced by the letter "e" with the acute accent (é) in page 1, lines 26 and 30. Although the word "resumé" is also acceptable with the letter "e" instead of the letter "e" with the acute accent (é), applicant is

advised to choose either convention and make the correction in order to keep the terminology consistent throughout the specification.

- o The reference number "110" should be changed to "100" in page 5, line 7, since "110" is used to designate the document, whereas "100" is used to designate the application.
- o The device "an optical drive interface" should be changed to "an optical disk drive interface" in page 12, line 22-23. Applicant is advised to make the correction in order to keep the terminology consistent between the drawing and the specification.
- o The phrase 'complete the "objective" section the performance review document ...' should presumably be read 'complete the "objective" section in the performance review document ...' in page 14, line 22.
- o The letter "a" in "active," the letter "s" in "server," and the letter "p" in "page" should be capitalized in page 22, line 27.
- o The word "an" should be changed to "and" in page 25, line 21.

- The explanation of what the acronym "dll" stands for should be stated after the first occurrence of the acronym "dll," which is in page 5, lines 29.
- The specification contains missing application number and filing date for the patent application incorporated by reference in page 10, line 30.
- The specification contains a program listing with more than 60 lines of code, which is submitted as part of the specification, must be positioned at the end of the description, but before the claims. See 37 CFR 1.96(b)(2)(ii).

Appropriate correction is required.

Claim Objections

6. **Claims 5, 6, 8, 9, and 12-18** are objected to because of the following informalities:

- **Claims 5 and 16** contain a typographical error: the letter "a" in "active," the letter "s" in "server," and the letter "p" in "page" should be capitalized.
- **Claim 6** contains a typographical error: the phrase "profile information associated for the user of the document" should presumably be read "profile information associated with the user of the document" in the sixth limitation.
- **Claim 8** contains a typographical error: the phrase "each namespace/solution matches ..." should presumably be read "each namespace/solution pair matches ..."
- **Claim 9** contains the following typographical errors:
    - There should be a semicolon (;) instead of a comma (,) at the end of second limitation.
    - There should be a comma (,) instead of a semicolon (;) to separate the "if" clause and the "replacing" clause.
- **Claims 12-18** contain a typographical error: there should be a hyphen (-) between the words "computer" and "readable" in the preamble.
- **Claim 18** contains a typographical error: the word "method" should be deleted in the preamble.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. **Claims 1-4, 6, 9, and 12-15** are rejected under 35 U.S.C. 102(b) as being anticipated by **Donohue et al. (US 5,987,480).**

As per **Claim 1**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality, comprising the steps of:

A. Attaching a schema to a document defining permissible data content, data type and data structure for the document *(see Column 8, Lines 25-54)*;

B. Structuring the document to associate the document with the schema *(see Column 10, Lines 10-17)*;

C. Associating a document solution with the document structure *(see Column 10, Lines 34-42)*;

D. Assembling a plurality of software components comprising one or more document solutions at a location remote from the document *(see Column 10, Lines 49-51)*;

E. Obtaining profile information associated with a user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

F. Generating a document solution tailored to the profile information associated with the user of the document *(see Column 11, Lines 21-25)*; and

G. Downloading the tailored document solution to the application for provision of functionality provided by the tailored document solution to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18).*

As per **Claim 2**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 1 above**, and further disclose that assembling the plurality of software components includes assembling the plurality of software components comprising one or more document solutions in a manifest of document solutions *(see Column 7, Lines 37-41)*; and

Prior to obtaining profile information associated with a user of the document, calling the manifest to request the tailored document solution, and passing an identification of the user of the document to the manifest with the request for the tailored document *(see Column 7, Lines 49-53 and 64-67; and Column 8, Lines 1-2).*

As per **Claim 3**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 2 above**, and further disclose that in response to the identification of the user of the document, calling a user information database from the manifest to obtain profile information associated with the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*; and

Generating a document solution tailored to the profile information associated with the user of the document includes selecting one or more document solution components from a plurality of document solution components based on the profile information *(see Column 10, Lines 60-65)*.

As per **Claim 4**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 3 above**, and further disclose that the method, prior to the step of downloading the software components to the application, comprising the steps of:

A. Determining whether the document solution associated with the document structure is present in a local library of software components *(see Figure 5, Element 110; and Column 12, Lines 58-60)*; and

B. If the plurality of software components is not present in the local library of software components, calling the manifest for obtaining the document solution *(see Column 7, Lines 49-53)*.

As per **Claim 6**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document, comprising the steps of:

A. Obtaining the document *(see Figure 3A, Element 48)*;

B. Determining whether the document references a document solution *(see Figure 3A, Element 54; and Column 10, Lines 37-42)*;

C. If the document references a document solution, calling a manifest of document solutions for the document solution *(see Column 7, Lines 49-53 and 64-67; and Column 8, Lines 1-2)*;

D. Passing an identification of a user of the document to the manifest of document solutions *(see Column 7, Lines 49-53 and 64-67; and Column 8, Lines 1-2)*;

E. At the manifest, calling a database of user information with the identification of the user for obtaining profile information for the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

F. Obtaining profile information associated with the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

G. At the manifest, generating a document solution tailored to the profile information associated with the user of the document *(see Column 11, Lines 21-25)*; and

H. Downloading the tailored document solution to the application for provision of functionality provided by the tailored document solution to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18).*

As per **Claim 9**, Donohue et al. disclose a method of managing a document solution downloaded by a software application for use with one or more documents, comprising:

A. Obtaining a document at the software application *(see Figure 3A, Element 48)*;

B. Determining whether the document contains a property identifying the document as being part of a document solution *(see Figure 3B, Elements 58 and 60; and Column 10, Lines 51-55)*;

C. If the document contains a property identifying the document as being part of a document solution, passing a solution directory for a document solution matching the property identifying the document as being part of a document solution *(see Figure 3B, Element 62; and Column 10, Lines 60-65)*; and

D. If the solution directory contains a document solution matching the property identifying the document as being part of a document solution, replacing the document solution contained in the solution directory with the document obtained at the software application *(see Figure 3B, Element 64; and Column 10, Lines 60-65).*

As per **Claim 12**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality, comprising the steps of:

A. Attaching a schema to a document defining permissible data content, data type and data structure for the document *(see Column 8, Lines 25-54)*;

B. Structuring the document to associate the document with the schema *(see Column 10, Lines 10-17)*;

C. Associating a document solution with the document structure *(see Column 10, Lines 34-42)*;

D. Assembling a plurality of software components comprising one or more document solutions at a location remote from the document *(see Column 10, Lines 49-51)*;

E. Obtaining profile information associated with a user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

F. Generating a document solution tailored to the profile information associated with the user of the document *(see Column 11, Lines 21-25)*; and

G. Downloading the tailored document solution to the application for provision of functionality provided by the tailored document solution to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18).*

As per **Claim 13**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 12 above**, and further disclose that assembling the plurality of software components includes assembling the plurality of software components comprising one or more document solutions in a manifest of document solutions *(see Column 7, Lines 37-41)*; and

Prior to obtaining profile information associated with a user of the document, calling the manifest to request the tailored document solution, and passing an identification of the user of the document to the manifest with the request for the tailored document *(see Column 7, Lines 49-53 and 64-67; and Column 8, Lines 1-2).*

As per **Claim 14**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 13 above**, and further disclose that in response to the identification of the user of the document, calling a user information database from the manifest to obtain profile information associated with the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*; and

Generating a document solution tailored to the profile information associated with the user of the document includes selecting one or more document solution components from a plurality of document solution components based on the profile information *(see Column 10, Lines 60-65)*.

As per **Claim 15**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 14 above**, and further disclose that the method, prior to the step of downloading the software components to the application, comprising the steps of:

A. Determining whether the document solution associated with the document structure is present in a local library of software components *(see Figure 5, Element 110; and Column 12, Lines 58-60)*; and

B. If the plurality of software components is not present in the local library of software components, calling the manifest for obtaining the document solution *(see Column 7, Lines 49-53).*

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. **Claims 5 and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Donohue et al. (US 5,987,480).**

As per **Claim 5**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 4 above**. However, Donohue et al. does not explicitly disclose that the manifest is an Active Server Page operative to call the user information database to obtain the profile information for the user of the document and to generate the tailored document solution by selecting one or more document solution components from a plurality of document solution components based on the profile information.

Nevertheless, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the manifest as an Active Server Page operative in the system of Donohue et al., since Donohue et al. already stores the manifest on a Web server containing script files *(see Column 6, Line 67; and Column 7, Lines 1-14 and 34-35)*. One would have been motivated to implement the manifest as an Active Server Page operative since ASP is a very well known Web programming scripting language that provides simplicity, speed, and security.

As per **Claim 16**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing updates or additions to application or document functionality **as in Claim 15 above**. However, Donohue et al. does not explicitly disclose that the manifest is an Active Server Page operative to call the user information database to obtain the profile information for the user of the document and to generate the tailored document solution by selecting one or more document solution components from a plurality of document solution components based on the profile information.

Nevertheless, it would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the manifest as an Active Server Page operative in the system of Donohue et al., since Donohue et al. already stores the manifest on a Web server containing script files *(see Column 6, Line 67; and Column 7, Lines 1-14 and 34-35)*. One would have been motivated to implement the manifest as an Active Server Page operative since ASP is a very well known Web programming scripting language that provides simplicity, speed, and security.

11. **Claims 7, 8, 10, 11, 17, and 18** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Donohue et al. (US 5,987,480)** in view of **Forbes et al. (US 6,381,742)**.

As per **Claim 7**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document **as in Claim 6 above**, and further disclose calling the location of the document solution identified by the document solution identification *(see Column 7, Lines 49-53)* and downloading the document solution identified by the document solution identification to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*.

However, Donohue et al. does not explicitly disclose that if the document does not reference a document solution, determining whether the document references a namespace associated with structure applied to the document; if the document references a namespace, calling a manifest collection, and determining whether the manifest collection contains a document solution identification associated with the document namespace; and if the manifest collection contains a document solution identification associated with the document namespace, obtaining a location of the document solution identified by the document solution identification.

In the same field of endeavor, Forbes et al. disclose a software package manager that uses a distribution unit containing components for a software package and a manifest file that describes the distribution unit to manage the installation, execution, and uninstallation of software packages on a computer. In the system of Forbes et al., the presence of a namespace XML tag in the manifest file causes the package manager to associate the files and components of the corresponding application in the code store data structure with the unique namespace specified in the tag *(see Column 14, Lines 20-24)*. When an application is executed, the package manager passes the associated namespace name to the computer's runtime environment so that any files and components installed in that namespace are visible to the application *(see Column 14, Lines 24-29)*. The manifest file is stored separately from the distribution unit *(see Column 14, Lines 49-52)*. The manifest file directs the package manager to the location of the distribution unit for the software application *(see Column 14, Lines 58-60)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate a namespace associated with the document in the system of Donohue et al., since Donohue et al. already utilizes a template that contains text and standard HTML tags *(see Figure 2; and Column 8, Lines 25-30)* for declaring namespaces and a data source storing content to be inserted in the templates *(see Figure 1, Element 12; and Column 7, Lines 37-39)*. One would have been motivated to incorporate a namespace associated with the document in order to assure that applications will function correctly even though identically named and having common components or files and that the applications will continue to function correctly irregardless of the number of applications using the same components or files, which may be installed on the computer *(see Column 14, Lines 42-48)*.

As per **Claim 8**, Donohue et al., as modified by Forbes et al., disclose a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document **as in Claim 7 above**. However, Donohue et al., and Forbes et al., do not explicitly disclose that the method further comprising populating the manifest collection with one or more namespace/solution pairs whereby each namespace/solution pair matches a document solution to a particular document namespace.

Nevertheless, it would have been obvious to one of ordinary skill in the art at the time the invention was made to populate the manifest collection with one or more namespace/solution pairs whereby each namespace/solution pair matches a document solution to a particular document namespace in the system of Donohue et al., since Donohue et al. is already populating other related data in the same manner where content from the data source are store in a container class as a pool of name/value pairs *(see Column 7, Lines 59-61)*. One would have been motivated to populate the manifest collection with one or more namespace/solution pairs whereby each namespace/solution pair matches a document solution to a particular document namespace in order to provide a quick and efficient lookup of one-to-one relationship data.

As per **Claim 10**, Donohue et al. disclose a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document, comprising the steps of:

A. Obtaining the document *(see Figure 3A, Element 48)*;

B. Calling the location of the document solution identified by the document solution identification *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*; and

C. Downloading the document solution identified by the document solution identification to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*.

However, Donohue et al. does not explicitly disclose that the method comprising the steps of:

A. Determining whether the document references a document namespace;

B. If the document references a document namespace, determining whether a manifest collection contains a document solution identification associated with the document namespace; and

C. If the manifest collection contains a document solution identification associated with the document namespace, obtaining a location of the document solution identified by the document solution identification.

In the same field of endeavor, Forbes et al. disclose a software package manager that uses a distribution unit containing components for a software package and a manifest file that describes the distribution unit to manage the installation, execution, and uninstallation of software packages on a computer. In the system of Forbes et al., the presence of a namespace XML tag in the manifest file causes the package manager to associate the files and components of the corresponding application in the code store data structure with the unique namespace specified in the tag *(see Column 14, Lines 20-24)*. When an application is executed, the package manager passes the associated namespace name to the computer's runtime environment so that any files and components installed in that namespace are visible to the application *(see Column 14, Lines 24-29)*. The manifest file is stored separately from the distribution unit *(see Column 14, Lines 49-52)*. The manifest file directs the package manager to the location of the distribution unit for the software application *(see Column 14, Lines 58-60)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate a namespace associated with the document in the system of Donohue et al., since Donohue et al. already utilizes a template that contains text and standard HTML tags *(see Figure 2; and Column 8, Lines 25-30)* for declaring namespaces and a data source storing content to be inserted in the templates *(see Figure 1, Element 12; and Column 7, Lines 37-39)*. One would have been motivated to incorporate a namespace associated with the document in order to assure that applications will function correctly even though identically named and having common components or files and that the applications will continue to function correctly irregardless of the number of applications using the same components or files, which may be installed on the computer *(see Column 14, Lines 42-48)*.

As per **Claim 11**, Donohue et al., as modified by Forbes et al., disclose a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document **as in Claim 10 above**, and Donohue et al. further disclose that the method, prior to downloading the document solution identified by the document solution identification to the document, comprising:

A. Passing an identification of a user of the document to a manifest of document solutions identified by the document solution identification as the location of the document solution *(see Column 7, Lines 37-41)*;

B. At the manifest, calling a database of user information with the identification of the user for obtaining profile information for the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

C. Obtaining profile information associated for the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

D. At the manifest, generating a document solution tailored to the profile information associated with the user of the document *(see Column 11, Lines 21-25)*; and

E. Whereby downloading the document solution identified by the document solution identification to the document includes downloading the tailored document solution to the document for providing the functionality of the tailored document solution to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*.

As per **Claim 17**, Donohue et al. disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document, comprising the steps of:

A. Obtaining the document *(see Figure 3A, Element 48)*;

B. Calling the location of the document solution identified by the document solution identification *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*; and

C. Downloading the document solution identified by the document solution identification to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18)*.

However, Donohue et al. does not explicitly disclose that the method comprising the steps of:

A. Determining whether the document references a document namespace;

B. If the document references a document namespace, determining whether a manifest collection contains a document solution identification associated with the document namespace; and

C. If the manifest collection contains a document solution identification associated with the document namespace, obtaining a location of the document solution identified by the document solution identification.

In the same field of endeavor, Forbes et al. disclose a software package manager that uses a distribution unit containing components for a software package and a manifest file that describes the distribution unit to manage the installation, execution, and uninstallation of software packages on a computer. In the system of Forbes et al., the presence of a namespace XML tag in the manifest file causes the package manager to associate the files and components of the corresponding application in the code store data structure with the unique namespace specified in the tag *(see Column 14, Lines 20-24)*. When an application is executed, the package manager passes the associated namespace name to the computer's runtime environment so that any files and components installed in that namespace are visible to the application *(see Column 14, Lines 24-29)*. The manifest file is stored separately from the distribution unit *(see Column 14, Lines 49-52)*. The manifest file directs the package manager to the location of the distribution unit for the software application *(see Column 14, Lines 58-60)*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate a namespace associated with the document in the system of Donohue et al., since Donohue et al. already utilizes a template that contains text and standard HTML tags *(see Figure 2; and Column 8, Lines 25-30)* for declaring namespaces and a data source storing content to be inserted in the templates *(see Figure 1, Element 12; and Column 7, Lines 37-39)*. One would have been motivated to incorporate a namespace associated with the document in order to assure that applications will function correctly even though identically named and having common components or files and that the applications will continue to function correctly irregardless of the number of applications using the same components or files, which may be installed on the computer *(see Column 14, Lines 42-48).*

As per **Claim 18**, Donohue et al., as modified by Forbes et al., disclose a computer-readable medium containing computer executable instructions which when executed by a computer perform a method of downloading software components from a remote source to a software application for providing a desired solution to a computer-generated document **as in Claim 17 above**, and Donohue et al. further disclose that the method, prior to downloading the document solution identified by the document solution identification to the document, comprising:

A. Passing an identification of a user of the document to a manifest of document solutions identified by the document solution identification as the location of the document solution *(see Column 7, Lines 37-41)*;

B. At the manifest, calling a database of user information with the identification of the user for obtaining profile information for the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

C. Obtaining profile information associated for the user of the document *(see Column 7, Lines 64-67; and Column 8, Lines 1-2)*;

D. At the manifest, generating a document solution tailored to the profile information associated with the user of the document *(see Column 11, Lines 21-25)*; and

E. Whereby downloading the document solution identified by the document solution identification to the document includes downloading the tailored document solution to the document for providing the functionality of the tailored document solution to the document *(see Column 7, Lines 6-7 and 25-33; and Column 11, Lines 16-18).*

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. **Rodov (US 6,697,837)** discloses a method of creating and storing a re-accessible, browser independent end user profile on the end user's computer, at least upon initial access, of an e-commerce website offering the purchase, downloading, and installation of software or information (data) therefrom, without requiring the user to repeatedly enter the profile information.

B. **Parthasarathy et al. (US 6,802,061)** disclose automatically downloading, verifying, installing, registering, and displaying computer software components from computer networks like Internet or an intranet.

C. **Murray et al. (US 6,874,143)** disclose systems and methods for providing software via a network by using software extensions.

D. **Lee et al. (US 6,880,129)** disclose a mechanism for generating namespaces in graphical user interface (GUI) page definitions.

E. **Glaser et al.** **(US 6,944,857)** disclose a system and method for updating an installation of an application program using a configuration of the application program corresponding to a particular user.

F. **Ben-Shaul et al.** **(US 6,976,090)** disclose a technique for content and application level distribution and customization of data and applications across the Internet using an integrated combination of origin servers and spatially distributed controlled edge servers to efficiently deliver content differentiated electronic content or data from content providers to various classes of consumers.

G. **Srivastava et al.** **(US 2002/0120685)** disclose methods and apparatus for providing information-based services from a plurality of diverse resources to one or more users.

H. **Mah et al.** **(US 2003/0014745)** disclose a method for updating a document module for use with an application program on a remote computer from a host computer.

I. **Lucovsky** **(US 2004/0199861)** discloses a schema-based documents service for Internet access to per-user document data, wherein access to data is based on each user's identity.

J. **Katano** **(US 2004/0201867)** discloses a system and method for providing updated help and solution information at a printing device.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, James W. Myhre, can be reached on 571-270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / QC
July 24, 2006

James W. Myhre
Supervisory Patent Examiner